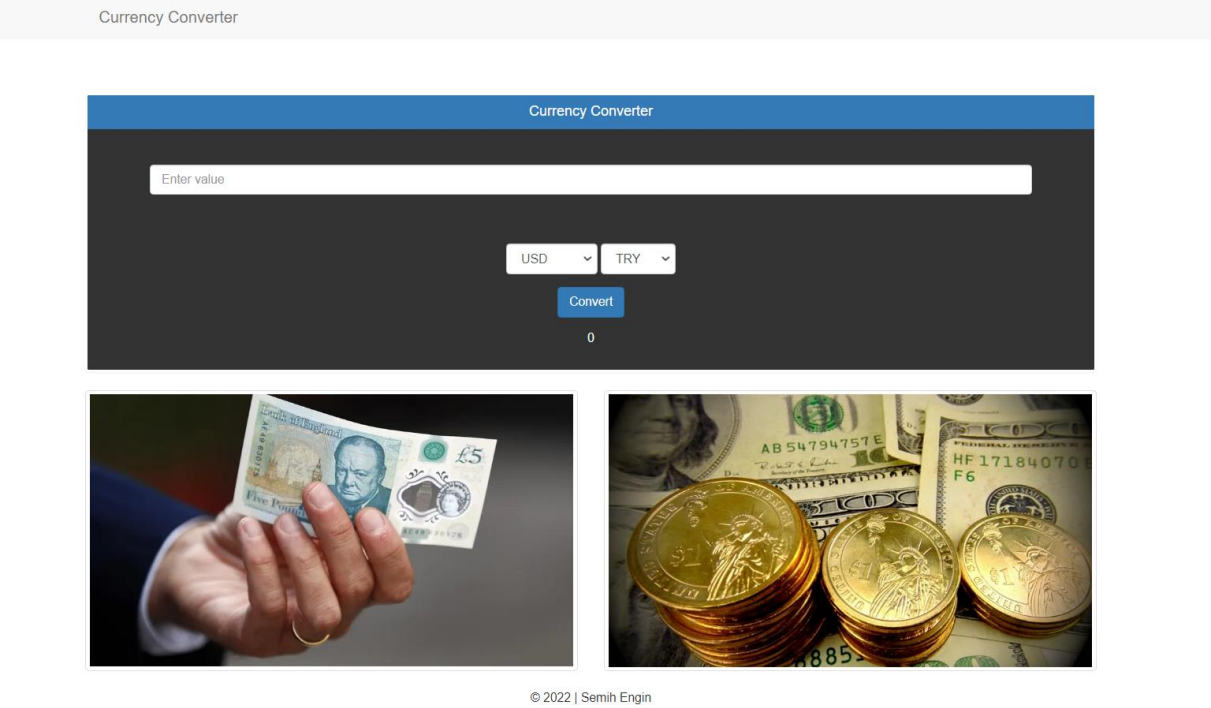
CURRENCY CONVERTER

My project is Currency Converter built with a simple microservice structure consisting of 4 servers, one of which is a gateway. I used Java SpringBoot for the backend part of the project because SpringBoot is a Java based framework used to build web applications and microservices. It's built on top of the Spring framework and takes care of many low-level details of building an application so I've used it for its advantages such as being able to focus on writing your business logic.



You can see the frontent of my website where I use microservices while writing the backend below. I used bootstrap while writing this and I created this image of the frontent by writing it with css where bootstrap was not enough.

```java
@SpringBootApplication
@RestController
@CrossOrigin(origins = "*",allowedHeaders = "*")
public class DemoApplication {

    4 usages
    private final RestTemplate restTemplate;
```

As you can see in the picture above, I made this project using the spring boot framework because it is popularly used in backend software. I used this spring boot because spring boot is a freamvork that allows writing powerful microservice structure. I used @CrossOrigin annotation CORS support needs to be turned on to use the service I created in the Spring Boot application Communicate with other applications.

```java
@GetMapping("/tl-dolar")
public String tlDolar(@RequestParam(value = "amount", defaultValue = "0") String amount,@RequestParam(value = "direction", defaultV
    String url = "http://localhost:5000/";
    if(direction.equals("1")){
        url += "usdToTry?amount="+ amount;
    }
    else{
        url += "tryToUsd?amount="+ amount;
    }
    return this.restTemplate.getForObject(url, String.class);
}

no usages
@GetMapping("/tl-euro")
public String tlEuro(@RequestParam(value = "amount", defaultValue = "0") String amount,@RequestParam(value = "direction", defaultVa
    String url = "http://localhost:4000/";
    if(direction.equals("1")){
        url += "euroToTry?amount="+ amount;
    }
    else{
        url += "tryToEuro?amount="+ amount;
    }
    return this.restTemplate.getForObject(url, String.class);
}

no usages
@GetMapping("/dolar-euro")
public String dolarEuro(@RequestParam(value = "amount", defaultValue = "0") String amount,@RequestParam(value = "direction", defaul
```

I created the edpoints with getmaping. here I created edpoints like tl-dollar eure-dollar. Using these edpoints, I send the required parameters to the edpoints from the frontend and show the necessary answers in the result section with the necessary answers. Then I coded 3 microservices for dollar-euro, tl_euro and tl_usd one by one and data is calculated thanks to these servers. In this way, if one server goes down, operations can continue on the other.

@RequestParam is value amount because we take the values we entered with this parameter and act according to the situation. For example, instead of typing server for tr_euro and euro_tr, we make a two-sided conversion with direction.equals on a single server with one condition.

```java
@GetMapping("/euroToTry")
public String euroToTry(@RequestParam(value = "amount", defaultValue = "0") String amount) {
    String url = "https://anyapi.io/api/v1/exchange/convert?apiKey=coicvh0u50gmhl6bn651h89q7diec06h875buf1r69g6fp1522m5ug&amount="+
    return this.restTemplate.getForObject(url, String.class);
}

no usages
@GetMapping("/tryToEuro")
public String tryToEuro(@RequestParam(value = "amount", defaultValue = "0") String amount) {
    String url = "https://anyapi.io/api/v1/exchange/convert?apiKey=coicvh0u50gmhl6bn651h89q7diec06h875buf1r69g6fp1522m5ug&amount="+
    return this.restTemplate.getForObject(url, String.class);
}
```

For example, to describe my tl_euro server, there are 2 edpoints, one from euro to tl and the other from euro to tl.there is an api url here. This apida takes the api key and the amount, we write it from what to what and convert it to base eur to what will be converted is try, namely tl.

In the amount part, the amount of the amount actually coming from the user to the gateway and directed here from the gatway is transferred here, thanks to this api I found from the internet, it tells how much is the euro and the value is returned from here.

```javascript
const toSelect = document.getElementById("to")
const convert = document.getElementById("convert")

convert.onclick = (e)=>{
    e.preventDefault()
    result.textContent = "..."
    const base = baseSelect.value // USD
    const to = toSelect.value // TRY
    const amount = amountInput.value
    if(base === to){
        result.textContent = amount;
    }
    else{
        let url = "http://localhost/"
        if(base === "TRY" && to === "USD"){
            url += "tl-dolar?amount="+amount
        }
        else if(base==="USD" && to === "TRY"){
            url += "tl-dolar?amount="+amount+"&direction=1"
```

I gave the urli as localhost because my gateway works on port 80 and when I don't write any port, it automatically works at 80. Then I added the url according to the condition of the base and to values and which edpoint it will be directed to under those conditions, together with the amount value.

```
        }
        axios.get(url)
        .then(res=>{
            console.log(res.data);
            result.textContent = res.data.converted
        })
        .catch(err=>{
            console.log(err);
            result.textContent = "Error"
        })
    }
```

I use Axios to send backend requests and return responses. In its simplest form, Axios is a JavaScript Libraries that allow us to easily perform HTTP operations in client applications. Below is part of if Using Axios based on the selected action contains a block that redirects to my API server.